FIG. 1

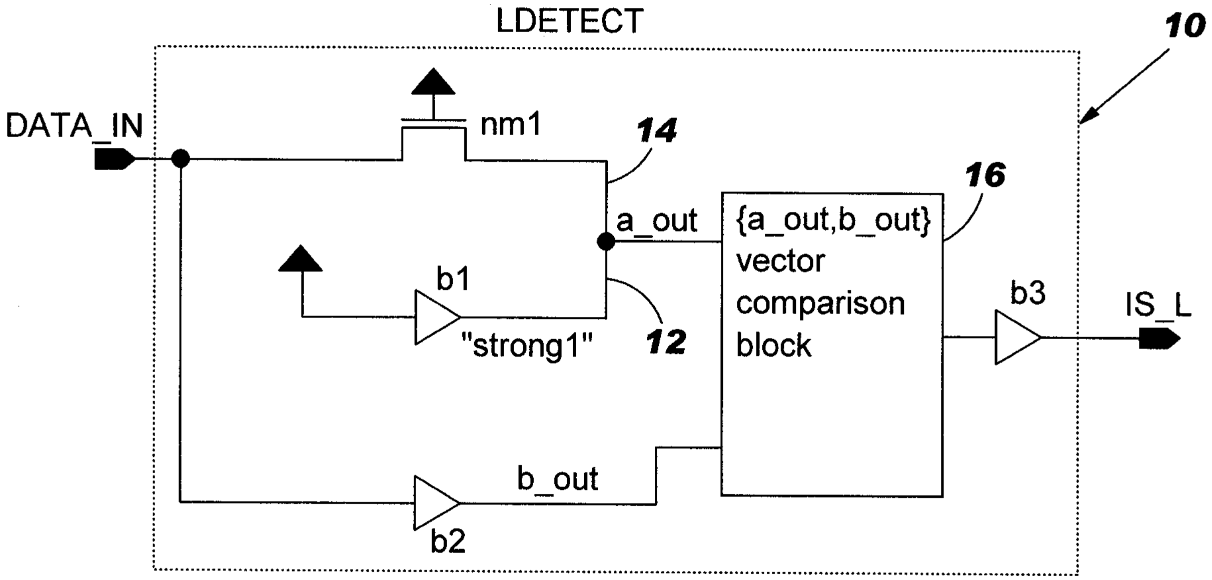

FIG. 2

20

DATA_IN	a_out	b_out	IS_L
"L"[1]	1	0	1
strong0	X	0	0
supply0	0	0	0
any strength logic 1	1	1	0
<=pullx	1	X	0
>=strongx	X	X	0
Hi-Z	1	X	0

[1] "L" is defined as any logic 0 of the following strengths: pull0, large0, weak0, medium0, or small0

FIG. 3

30

Verilog HDL example code for "LDETECT" module

```
module LDETECT (IS_L, DATA_IN);
  output IS_L;
  input DATA_IN;

  nmos   nm1  (a_out,DATA_IN,1'b1);
  buf    b1   (a_out,1'b1);    //drive & compare with Strong1

  buf    b2   (b_out,DATA_IN);

  wire   preIS_L = ({a_out,b_out}  === 2'b10);
  buf    b3    (IS_L, preIS_L);

endmodule
```

FIG. 4

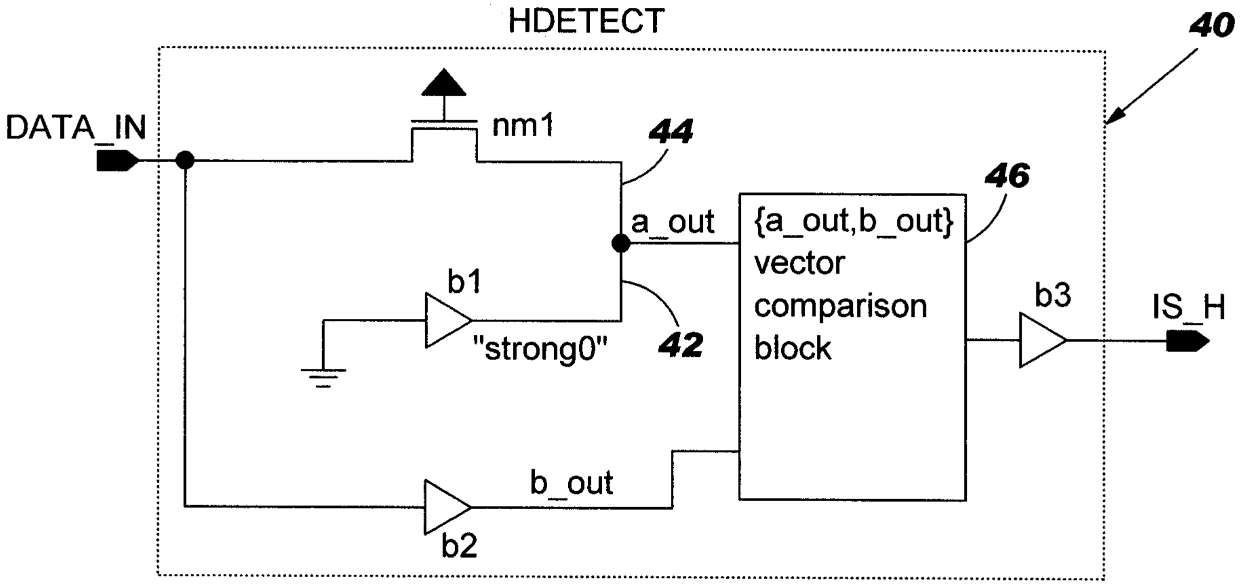

FIG. 5

50

DATA_IN	a_out	b_out	IS_H
any strength logic 0	0	0	0
"H"[1]	0	1	1
strong1	X	1	0
supply1	1	1	0
<=pullx	0	X	0
>=strongx	X	X	0
Hi-Z	0	X	0

[1] "H" is defined as any logic 1 of the following strengths: pull1, large1, weak1, medium1, or small1

FIG. 6

60

Verilog HDL example code for "HDETECT" module

```
module HDETECT (IS_H, DATA_IN);
  output IS_H;
  input DATA_IN;

  nmos   nm1  (a_out,DATA_IN,1'b1);
  buf    b1   (a_out,1'b0);     //drive & compare with Strong0

  buf    b2   (b_out,DATA_IN);

  wire   preIS_H = ({a_out,b_out}  === 2'b01);
  buf    b3    (IS_H, preIS_H);

endmodule
```

FIG. 7

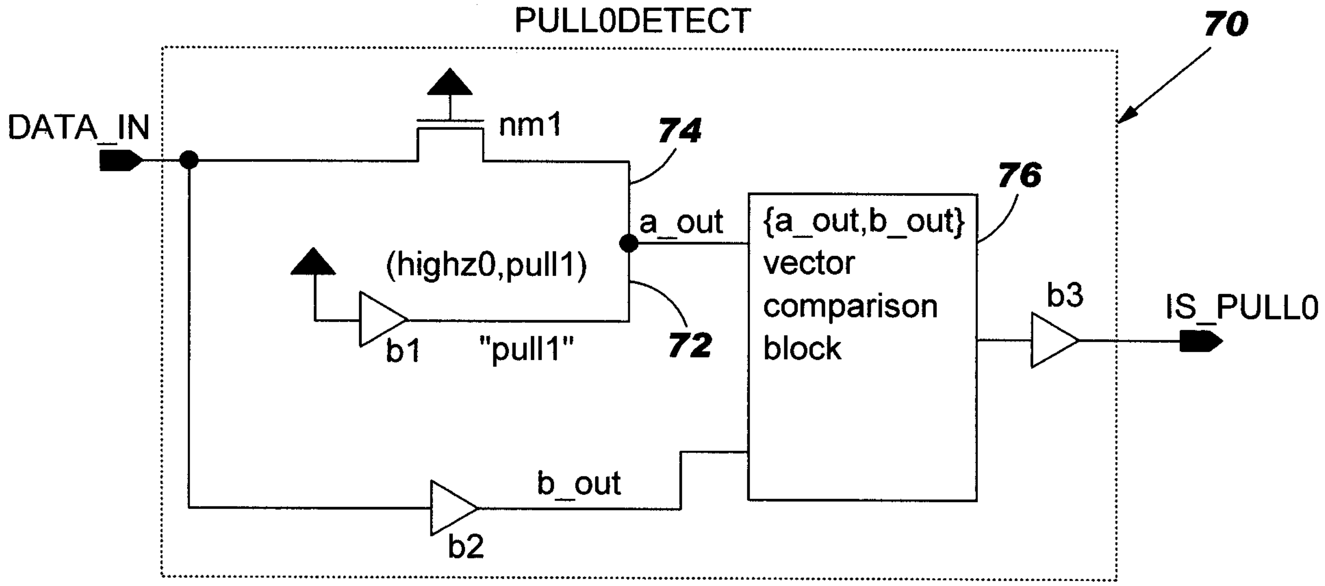

FIG. 8

80

DATA_IN	a_out	b_out	IS_PULL0
<=large0	1	0	0
pull0	X	0	1
>=strong0	0	0	0
any strength logic 1	1	1	0
<=largex	1	X	0
>=pullx	X	X	0
Hi-Z	1	X	0

FIG. 9

90

Verilog HDL example code for "PULL0DETECT" module

```
module PULL0DETECT (IS_PULL0, DATA_IN);
  output IS_PULL0;
  input DATA_IN;

  nmos   nm1  (a_out,DATA_IN,1'b1);
  //drive & compare with pull1
  buf      (highz0,pull1) b1     (a_out,1'b1);

  buf      b2     (b_out,DATA_IN);

  wire     preIS_PULL0 = ({a_out,b_out}  === 2'bx0);
  buf      b3      (IS_PULL0, preIS_PULL0);

endmodule
```

FIG. 10

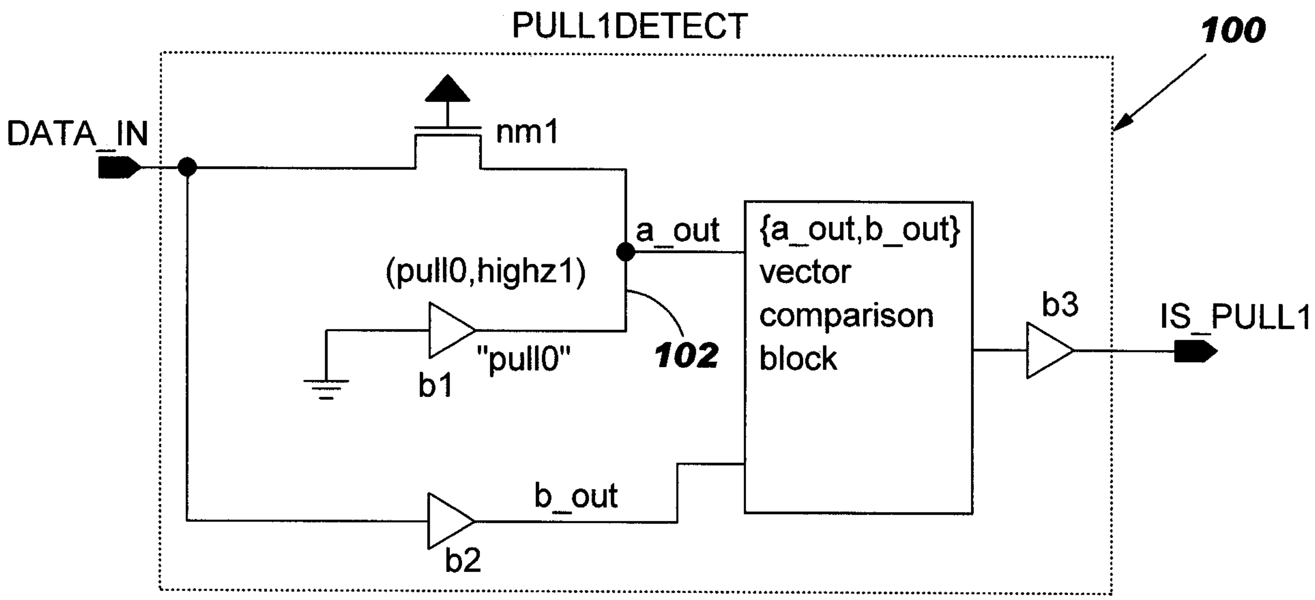

FIG. 11

110

DATA_IN	a_out	b_out	IS_PULL1
any strength logic 0	0	0	0
<=large1	0	1	0
pull1	X	1	1
>=strong1	1	1	0
<=largex	0	X	0
>=pullx	X	X	0
Hi-Z	0	X	0

FIG. 12

120

Verilog HDL example code for "PULL1DETECT" module

```
module PULL1DETECT (IS_PULL1, DATA_IN);
  output IS_PULL1;
  input DATA_IN;

  nmos   nm1  (a_out,DATA_IN,1'b1);
  //drive & compare with pull0
  buf      (pull0,highz1) b1     (a_out,1'b0);

  buf      b2     (b_out,DATA_IN);

  wire     preIS_PULL1 = ({a_out,b_out}  === 2'bx1);
  buf      b3      (IS_PULL1, preIS_PULL1);

endmodule
```

FIG. 13

Example IO: Driver Truth Table

130

	A	TS	DI	PAD	PADN
a	-	0	-	L[1]	L[1]
b	-	-	0	L[1]	L[1]
c	-	1	1	A	not-A

[1] "L" in this example refers to a pull strength logic0. Also known as "pull0".

FIG. 14

Example IO: Receiver Truth Table

140

	PAD	PADN	RG	Z
d	0	1	1	0
e	1	0	1	1
f	1	1	1	X
g	0	0	1	X
h	L[1]	L[1]	1	0
i	L[1]	0	1	X
j	L[1]	1	1	X
k	0	L[1]	1	X
l	1	L[1]	1	X
m	-	-	0	0

[1] "L" in this example refers to pull strength logic 0. Also known as "pull0".

FIG. 15

Verilog HDL example code for modeling "Example IO" truth tables above

150

```
module Example_IO (PAD,PADN,Z,A,TS,DI,RG);
      inout  PAD,  PADN;
      output  Z;
      input  A,  TS,  DI,  RG;
      tri    outBuf, outBufN;
      wire  bufCtrl, isPadPull0, isPadNPull0, muxSel1, muxOut1, muxOut2;

      //Driver section
      and   a1   (bufCtrl,TS,DI);
      not   n1   (A_,A);

      bufif1       bf1    (outBuf, A, bufCtrl);
      bufif1       bf2    (outBufN, A_,bufCtrl);
      pulldown  d1     (outBuf);
      pulldown  d2     (outBufN);
      nmos         nm1  (PAD, outBuf, 1'b1);
      nmos         nm2  (PADN, outBufN,1'b1);

      //Receiver section
      PULL0DETECT  pd1  (isPadPull0, PAD);     //Reference Fig. 9 for PULL0DETECT
                                                                module definition
      PULL0DETECT  pd2  (isPadNPull0,PADN); //Reference Fig. 9 for PULL0DETECT
                                                                 module definition
      xnor      xn1    (muxSel1,PAD,PADN);
      //MUX21 port order: Mux_Out, Data_Sel, D0, D1
      MUX21*  mx1  (muxOut1,muxSel1,PAD,1'bx);
      //MUX41 port order: Mux_Out, Data_Sel(MSB), Data-Sel(LSB), D00, D01, D10, D11
      MUX41*  mx2  (muxOut2,isPadPull0,isPadNPull0,muxOut1,1'bx,1'bx,1'b0);

      and        a2    (Z,muxOut2,RG);
      endmodule
```

*NOTE that the "MUX21" and "MUX41" instances in the Verilog code above are not shown modeled here for this example, but are industry standard 2:1 and 4:1 Multiplexers, respectively. They are not standard "built-in" Verilog primitives, but are easily implemented in Verilog either via behavioral code or by UDP primitive tables.